

Nama : Nazwa Audina
NPM : 51422238
Kelas : 2IA11
Mata Kuliah : Pemrograman Berbasis Objek

Rangkuman Materi Pertemuan 1-6

Pengenalan PBO (Pemrograman Berbasis Objek)

- Pemrograman berorientasi objek (OOP) adalah paradigma pemrograman yang menggunakan objek sebagai metode pembuatan program untuk menyelesaikan masalah yang kompleks. Objek memiliki atribut, karakter, dan kadang-kadang kondisi. OOP dikembangkan dari bahasa C dan Pascal pada tahun 1960. Pendekatan ini fokus pada objek yang dapat memecahkan masalah, bukan pada cara memecahkan masalah tersebut. OOP efektif ketika digabungkan dengan OOAD.

Pemrograman Berbasis Objek:

- Menggabungkan fungsi dan data dalam kelas atau objek.
- Mencakup konsep Encapsulation, Inheritance, dan Polymorphism.
- Struktur program ringkas dengan menggunakan objek dan kelas.
- Kode program sangat reusable.
- Efektif untuk menyelesaikan masalah besar karena memisahkan kode program menjadi kelompok-kelompok kecil (class).
- Awalnya sulit karena memerlukan pembuatan kelas.
- Eksekusi lebih cepat karena objek dieksekusi bersamaan.

Pemrograman Terstruktur:

- Memecah program dalam fungsi dan data.
- Mencakup konsep Sequence, Selection, dan Repetition.
- Struktur program rumit karena berupa urutan proses dan fungsi-fungsi.
- Re-use kode program kurang.

- Tidak cocok untuk menyelesaikan masalah yang rumit, karena sulit menemukan solusi permasalahan ketika terjadi error.
- Mudah diawal, namun kompleks diproses selanjutnya.
- Eksekusi lebih lambat karena setiap perintah dikerjakan berurutan.

➤ **Class**

adalah cetak biru atau blueprint dari sebuah objek. Class digunakan untuk membuat kerangka dasar suatu objek. Untuk memberi analogi, class dapat diibaratkan dengan kendaraan. Kendaraan memiliki ciri-ciri seperti merek, mesin, ban, dan lainnya yang menandakan sebuah benda sebagai kendaraan. Selain itu, kendaraan juga dapat melakukan tindakan seperti menghidupkan atau mematikan mesin.

➤ **Objek**

adalah entitas yang memiliki atribut, karakter, dan kadang kala kondisi. Objek merepresentasikan sesuatu dalam kehidupan nyata, seperti siswa, merek dagang, atau visualisasi seperti bentuk huruf. Objek mirip dengan rekaman (record) dalam sistem berkas, seperti rekaman karyawan. Sebuah objek didefinisikan oleh namanya atau kata benda, dan memiliki atribut serta metode.

Inheritance adalah sebuah mekanisme untuk "menggunakan kembali" (reusability) kode, di mana sebuah class baru dapat dibuat dari class yang sudah ada dengan menambahkan atau memodifikasi fasilitasnya.

Dalam inheritance, konstruktor tidak secara otomatis diwariskan ke subclass, kecuali jika digunakan perintah 'super'.

Ada dua jenis pewarisan:

- ❖ **Pewarisan tunggal (single inheritance):** Pewarisan dari satu class induk.
- ❖ **Pewarisan ganda (multiple inheritance):** Pewarisan dari dua atau lebih class induk. Namun, Java tidak mendukung multiple inheritance.

Class yang memberikan warisan disebut superclass atau parent class, sedangkan class yang menerima warisan (subclass atau derived class) disebut subclass.

Untuk menerapkan inheritance dalam Java, digunakan kata kunci "extends". Kata kunci "super" digunakan oleh subclass untuk memanggil konstruktor, atribut, dan metode yang ada pada superclassnya.

Encapsulation adalah cara untuk melindungi atribut atau metode tertentu dari sebuah kelas agar tidak dapat diakses atau dimodifikasi sembarangan oleh bagian program lainnya.

Cara melindungi data adalah dengan menggunakan access modifiers, yaitu:

- **Default** : Atribut atau metode dapat diakses oleh kelas dalam package yang sama.
- **Public** : Atribut atau metode dapat diakses dari mana saja.
- **Protected** : Atribut atau metode hanya dapat diakses oleh kelas dalam package yang sama atau oleh subclass dari kelas tersebut.
- **Private** : Atribut atau metode hanya dapat diakses oleh kelas itu sendiri.

Polymorphism:

- **Overloading** memungkinkan sebuah class memiliki beberapa metode dengan nama yang sama tetapi dengan parameter yang berbeda.
- **Overriding** adalah kemampuan sebuah subclass untuk memodifikasi metode dari superclassnya, dengan mendefinisikan kembali metode yang sudah ada dalam superclass. Ini dilakukan dengan menggunakan nama yang sama dan parameter yang sama atau kompatibel.